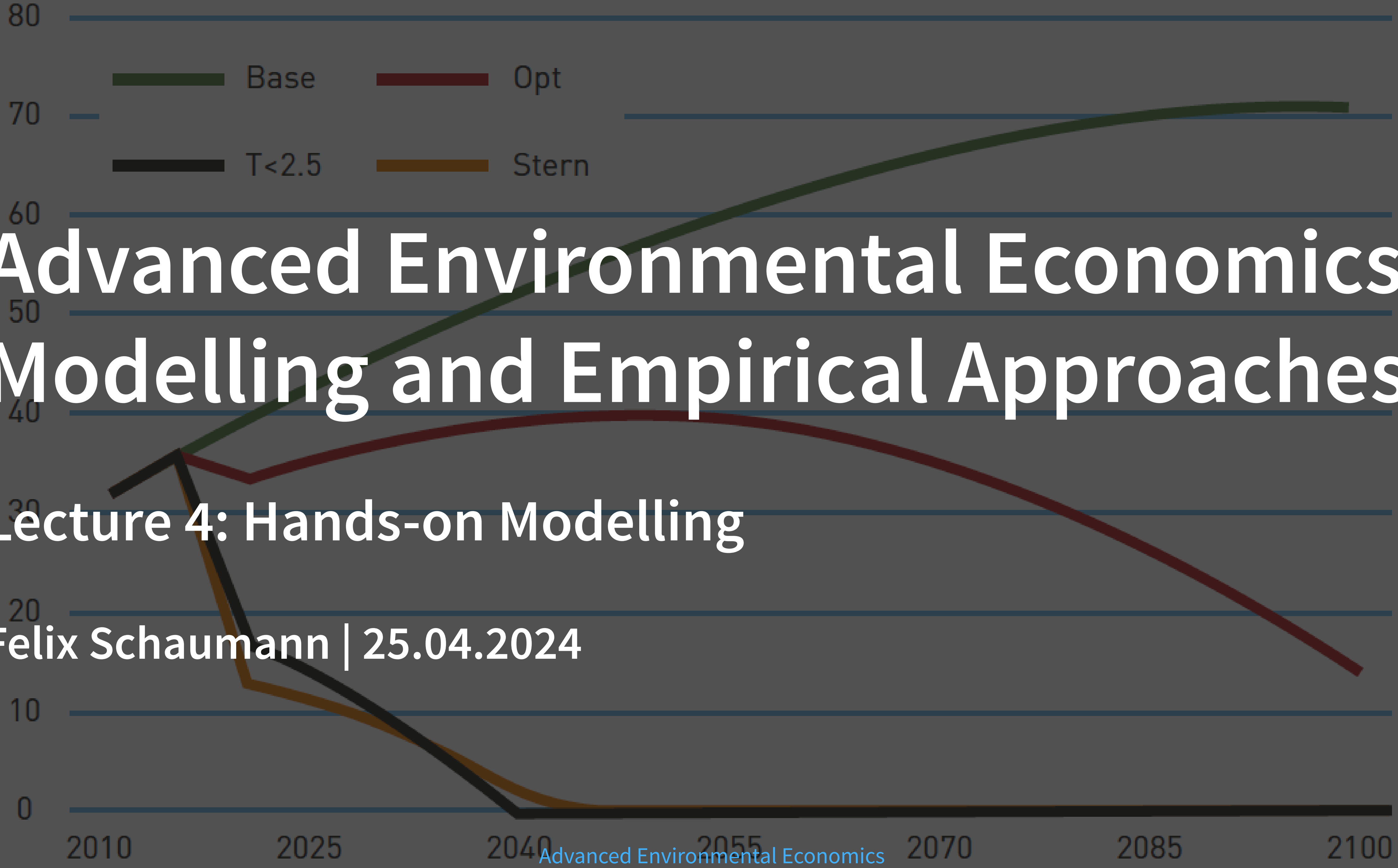
Advanced Environmental Economics: Modelling and Empirical Approaches

Lecture 4: Hands-on Modelling

Felix Schaumann | 25.04.2024



Recap

What have we done so far?

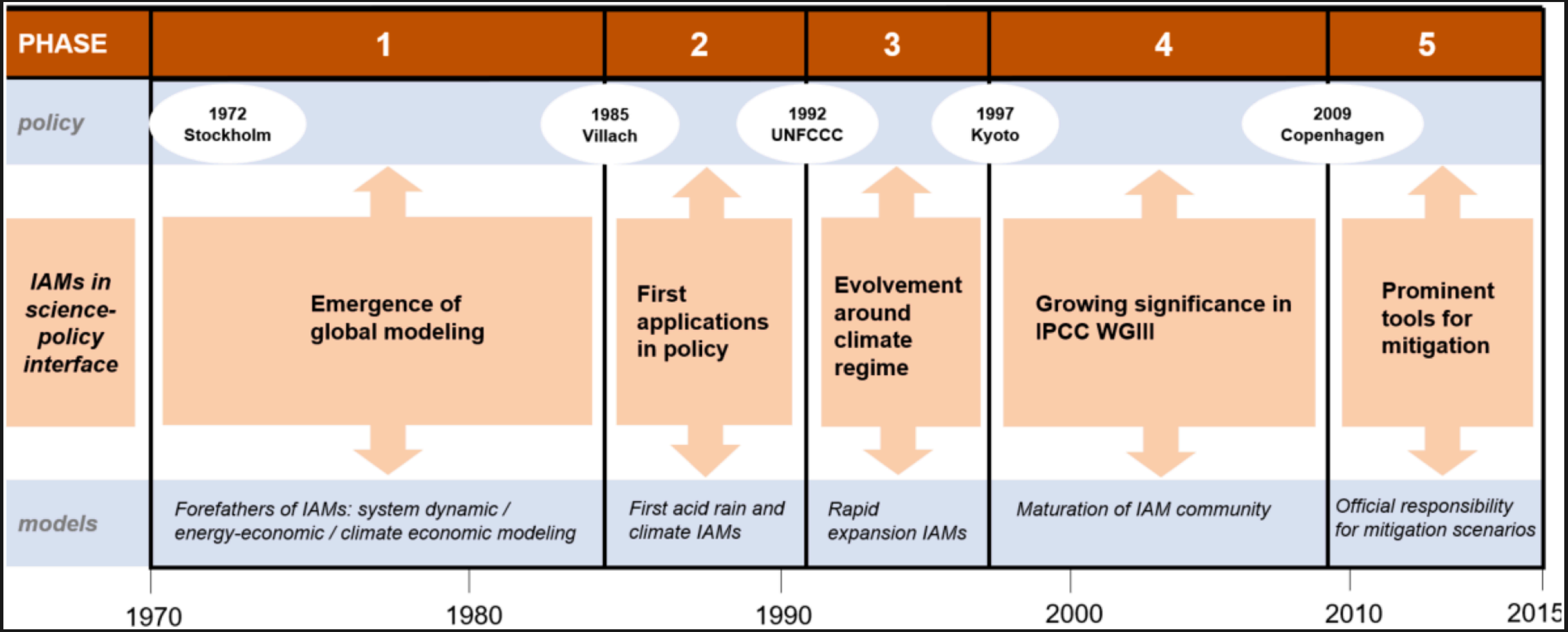
1. Introduction
2. Modelling and the economics of climate change
3. Empirical Approaches

- **Any questions before we move on?**

This lecture

1. IAMs: a brief history
2. How to code IAMs?
3. How to run an IAM in Mimi

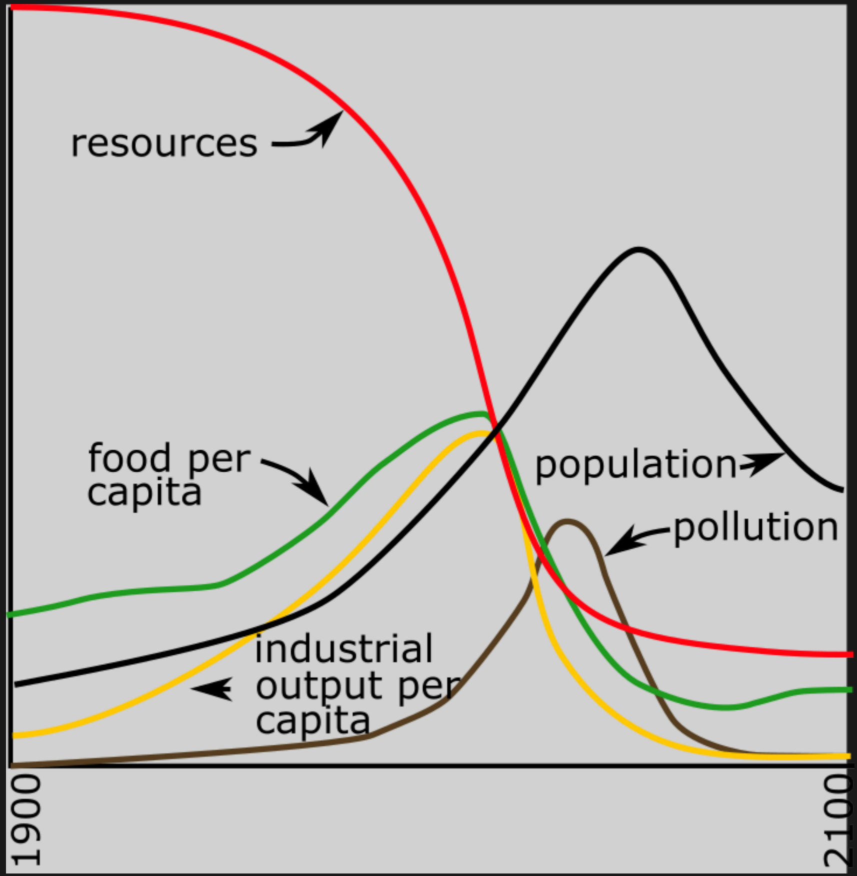
1. IAMs: a brief history



van Beek et al. (2020)

1972: Arguably the first IAM

- *The Limits to Growth* (Meadows et al., 1972)
- First report to the Club of Rome
- World3 simulation model
- birth of *system dynamics*
- similar: Bariloche model (Herrera et al., 1976)



- critiqued by Nordhaus (1973)
- controversial within economics
- influential for environmental movement

IPCC definition of **integrated assessment**:
"A method of analysis that combines results and models […] in a consistent framework to evaluate the status and the consequences of environmental change and the policy responses to it." (Allwood et al., 2014, p. 1264)

Nordhaus: integrating climate change into economics

TABLE 1. CONTROL STRATEGIES

1. REDUCE EMISSIONS:
   A. REDUCE DEMAND*
   B. SUBSTITUTION IN SUPPLY*
2. NEGATE DAMAGES
   A. MIX INTO OCEANS
   B. OTHER OFFSETTING EFFECTS (PARTICULATES, PAINT, BAND-AIDS)
3. CLEAN UP EX-POST
   A. REMOVE FROM AIR
   B. GROW TREES
4. NATURES WAY AND PRAY
   DO NOTHING (RULED OUT)

*CONSIDERED IN MODEL

Nordhaus (1975, IIASA)

Fig. 2. Greenhouse-gas concentrations. Concentrations of GHGs (including CO_2 and CFCs, in CO_2-equivalent basis) are estimated to double from preindustrial levels around 2050 for the uncontrolled path (+). The optimal path (□) shows a small reduction in concentrations, whereas emissions stabilization (•) leads to continued increases. Climate stabilization (——) would require declining concentrations.

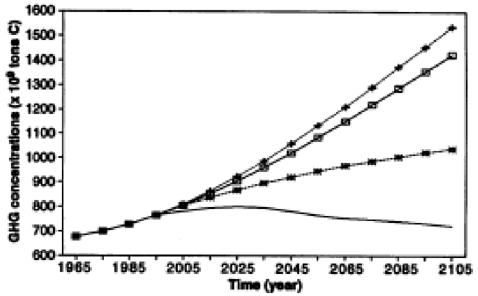


Fig. 3. Projected global mean temperature. According to the DICE model, global mean temperature with no controls (+) is projected to increase 3°C above 1900 levels by 2085. The optimal policy (□) and emissions stabilization (•) would involve only a small reduction in global warming. The maximum feasible policy is climate stabilization (——), which shows significant warming because of the commitment in the current buildup of GHGs.

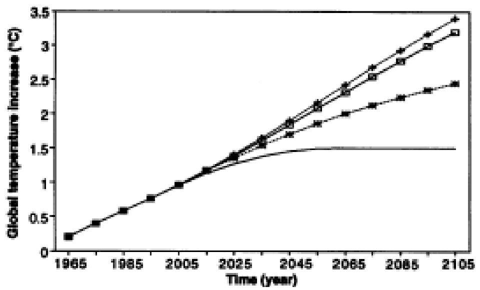
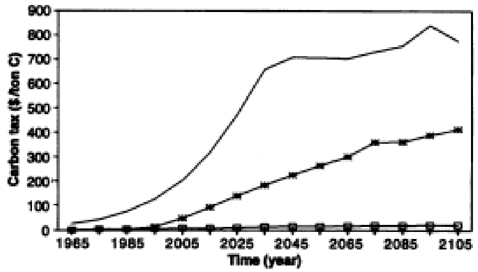


Fig. 4. Carbon taxes in different policies to reduce GHGs. Carbon taxes are a good index of the stringency of policies to slow global warming. A carbon tax would penalize production and consumption of fossil fuels and CFCs and encourage afforestation. Calculations indicate that the optimal carbon tax (□) rises from around \$5 to about \$20 per ton of carbon in 2100. Emissions stabilization (•) and climate stabilization (——) imply sharply rising carbon taxes that would raise coal prices severalfold. Carbon tax given in 1989 U.S. dollars.



Nordhaus (1992, Science)

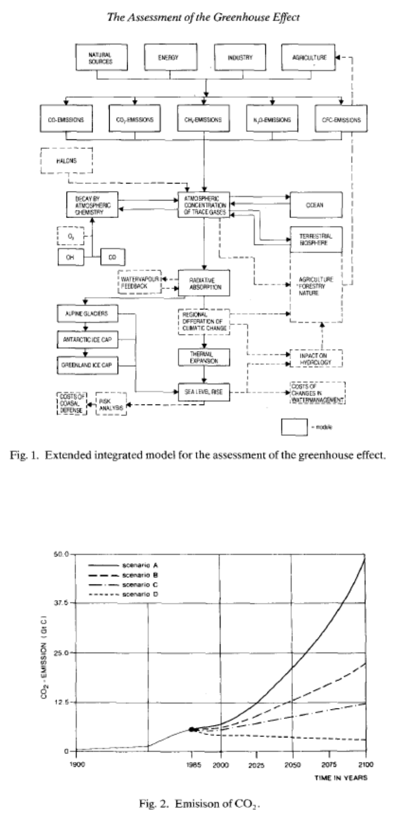
The Prize in Economic Sciences

William D. Nordhaus

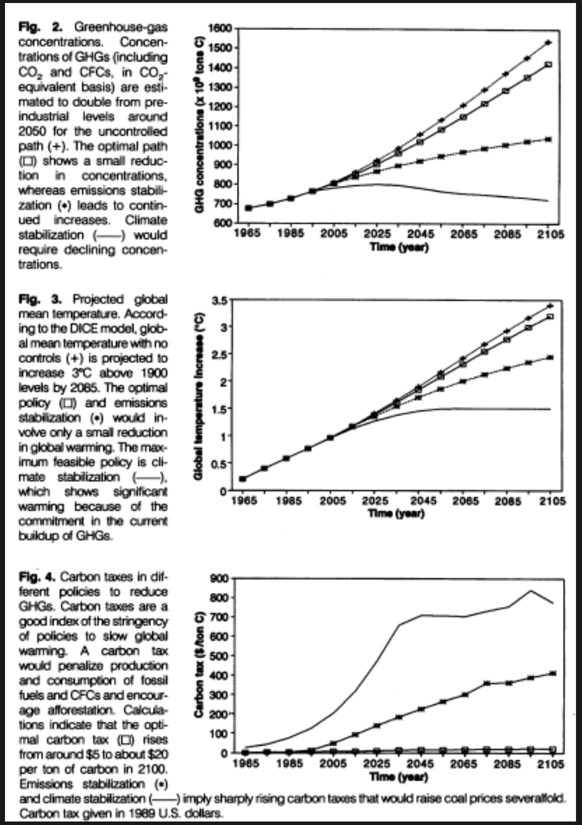
"for integrating climate change into long-run macroeconomic analysis"

© Nobel Media AB. Photo: A. Mahmoud

1990s: The emerging landscape of IAMs



- 1990: first assessment report by the IPCC
- 1992: Rio Earth Summit
- establishment UNFCCC
- new climate scenarios used in IPCC reports
- both DP-IAMs and CB-IAMs relevant to global climate policy discourse



Rotmans et al. (1990)

Nordhaus (1992)

- **detailed-process IAMs** (DP-IAMs)

- **cost-benefit IAMs** (CB-IAMs)

DICE: from optimal temperature to the SCC

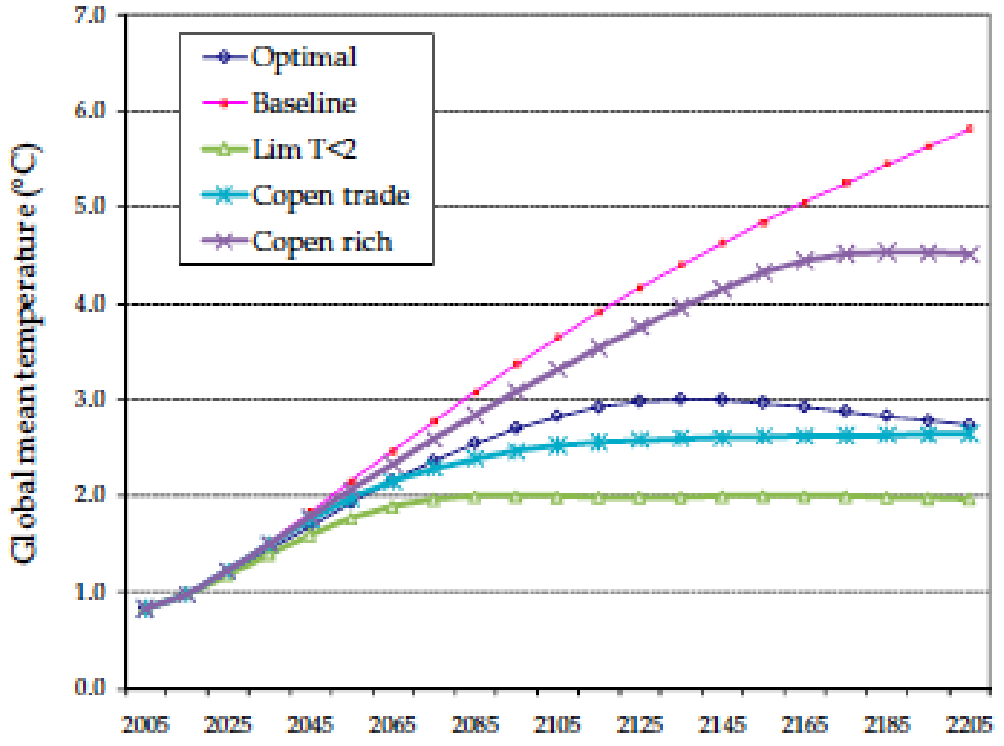


Fig. 3. Global temperature increase (°C from 1900) under alternative policies. Copen, Copenhagen.

Nordhaus (2010)

Table 1. Global Social Cost of Carbon under Different Assumptions

Scenario	2015	2020	2025	2030	2050
Base parameters:					
Baseline*	18.6	22.1	26.2	30.6	53.1
Optimal controls†	17.7	21.2	25.0	29.3	51.5
2°C limit damage function:					
Maximum†	47.6	60.1	75.5	94.4	216.4
Max of average†	25.0	30.6	37.1	44.7	87.9
Stern Review discounting:					
Uncalibrated*	89.8	103.7	117.4	131.3	190.0
Calibrated*	20.7	25.0	30.1	35.9	66.9
Alternative high discount*	6.4	7.7	9.2	10.9	19.6

Note.—The social cost of carbon is measured in 2005 international US dollars. The years at the top refer to the date at which emissions take place. Therefore, $18.6 is the cost of emissions in 2015 in terms of consumption in 2015.

* Calculation along the reference path with current policy.

† Calculation along the optimized emissions path.

Nordhaus (2014)

META: the effect of tipping points on the SCC

- SC-IAM
  → explicitly focused on SCC
- no optimisation
  → assumed baseline path
- How do climate tipping points affect the SCC?

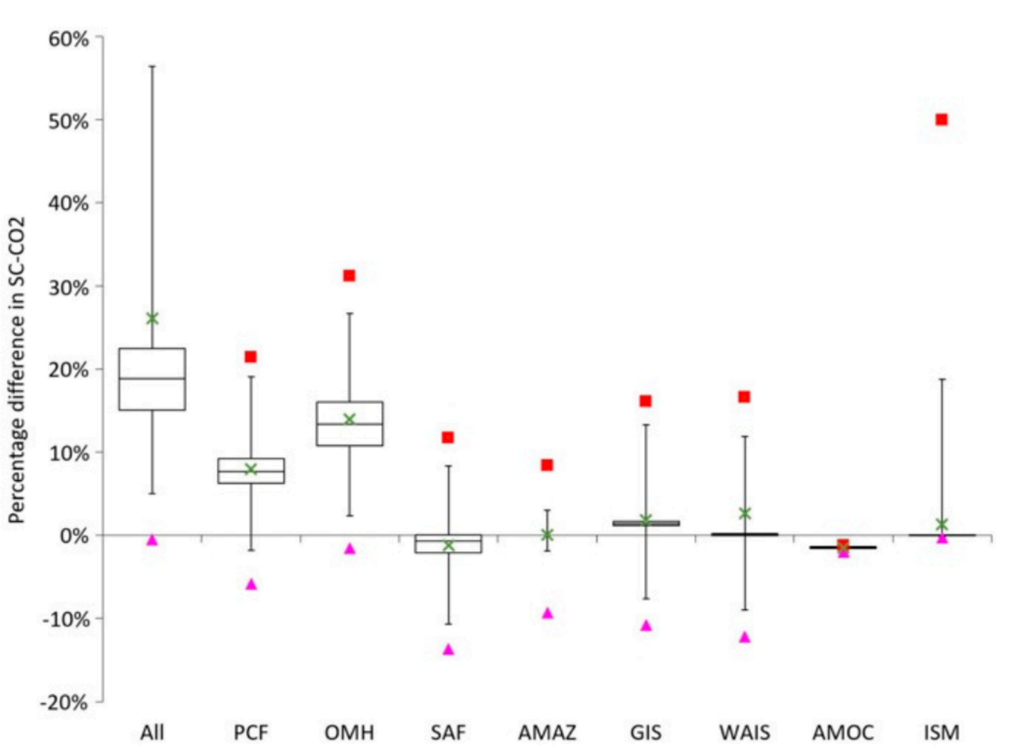


Fig. 1. The percentage change in the SCC due to tipping points collectively and individually. Boxes show medians and interquartile ranges, whiskers show 95% CIs, crosses mark the average changes (0.1% trimmed), triangles mark the 0.5 percentiles, and squares mark the 99.5 percentiles. The *y* axis is truncated. Specification comprises RCP4.5-SSP2 emissions and GDP/population growth, Hope and Schaefer PCF, Whiteman et al. beta OMH, and IPSL AMOC hosing. Monte Carlo sample size is 10,000.

Dietz et al. (2021, PNAS)

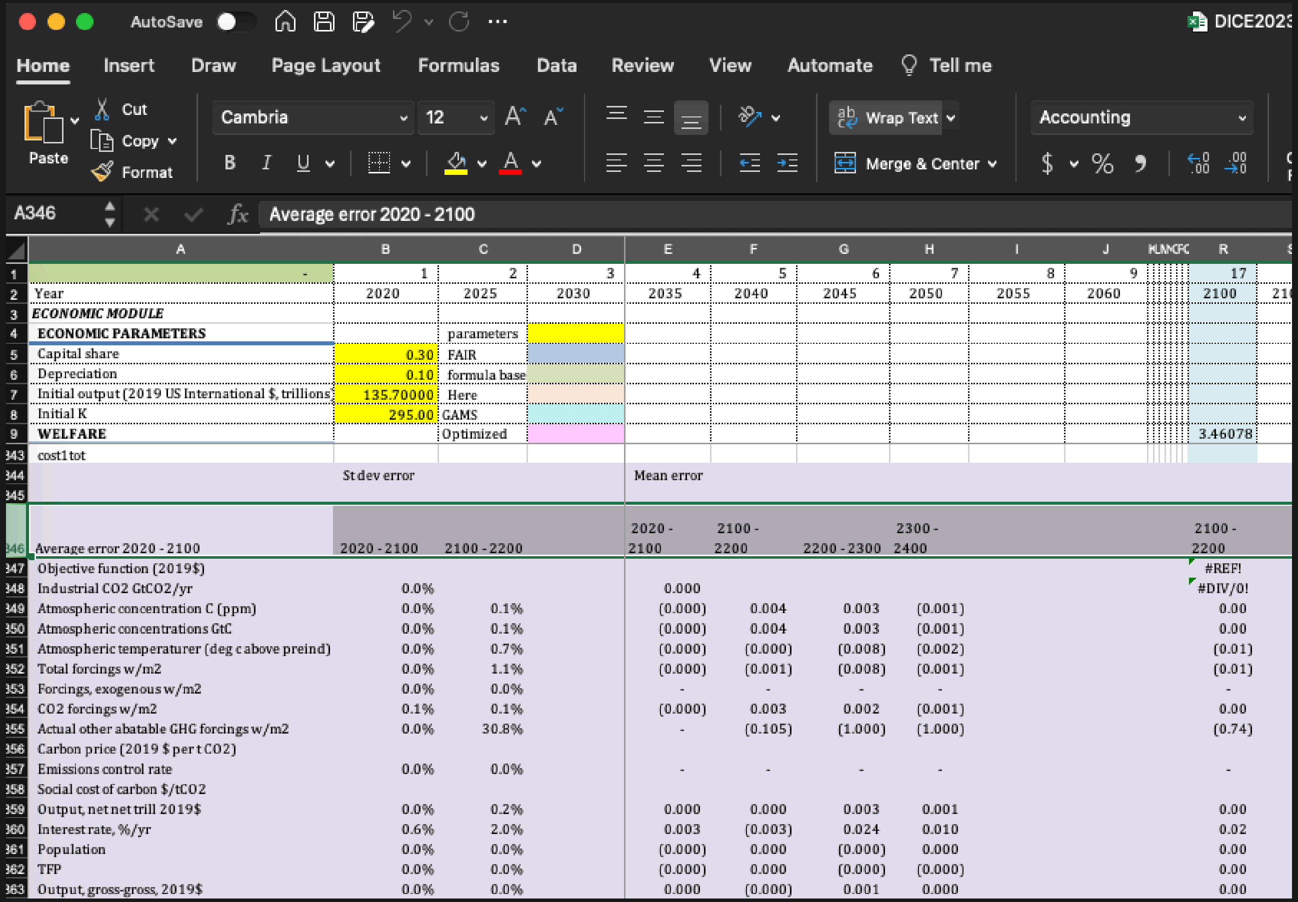
2. How to code IAMs?

Generally, one uses programming tools. The most common one…

src > META-2021 > components > AMOC_emulator.jl > run_timestep

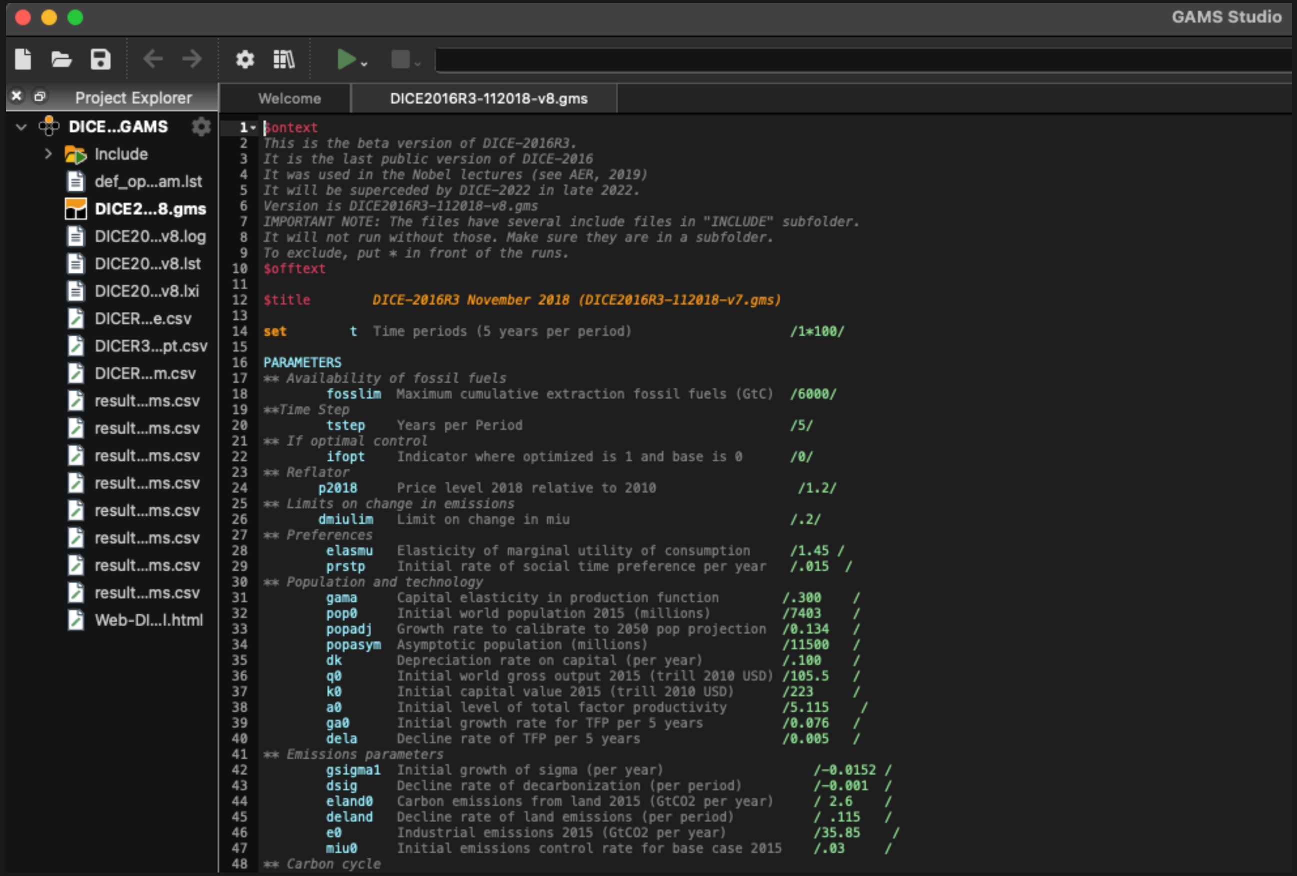
```
@defcomp AMOC_emulator begin
    country = Index()

    # Variables
    I_AMOC = Variable{Bool}(index=[time])
    AMOC_decrease = Variable(index=[time], unit="Sv") # AMOC decrease (measured in Sv) starting in t=0
    deltaT_country_AMOC = Variable(index=[time, country], unit="degC")
    T_country_AMOC = Variable(index=[time, country], unit="degC")
    CO2_AMOC = Variable(index=[time], unit="GtCO2")
    cum_CO2_AMOC = Variable(index=[time], unit="GtCO2")

    # Parameters
    T_AT = Parameter(index=[time], unit="degC")
    f_AMOC = Parameter(index=[time])

    beta_AMOC = Parameter() # coefficient representing the sensitivity of AMOC to temperature
    DecreaseOffset = Parameter() # offset for AMOC decrease by 2010 in Sv (with how much weakening AMOC starts in 2010)
    eta_AMOC = Parameter() # coefficient representing the sensitivity of the carbon drawdown to AMOC decrease
    tau_AMOC = Variable(index=[country]) # coefficient representing the sensitivity of the temperature change to AMOC decrease
    I_tau = Parameter(default=1) # indicator variable for gamma_AMOC
    q_AMOC = Parameter() # coefficient indicating the relative amount of AMOC slowdown in relative calibration

    #delta_AMOC = Parameter(unit="", default=35)
    max_deltaT_country_AMOC = Parameter(index=[country], unit="degC")

    T_country_base = Parameter(index=[time, country], unit="degC")

    T_AT_2010 = Parameter(default=0.854, unit="degC")
    AMOC_2010 = Parameter(default=18.0, unit="Sv")

    function run_timestep(pp, vv, dd, tt)

        # AMOC decrease (measured in Sv) starting in t=0 and assumed to be linear
        # this variable could also be directly calibrated to ESM output
        vv.AMOC_decrease[tt] = pp.DecreaseOffset + pp.beta_AMOC * (pp.T_AT[tt] - pp.T_AT_2010)
        vv.I_AMOC[tt] = 0

        if is_first(tt)

            # calculate CO2 release depending on AMOC strength
            # while not imposing a maximum CO2 release
            vv.CO2_AMOC[tt] = pp.eta_AMOC * vv.AMOC_decrease[tt]
            vv.cum_CO2_AMOC[tt] = vv.CO2_AMOC[tt]

            for cc in dd.country
                vv.tau_AMOC[cc] = pp.I_tau * pp.max_deltaT_country_AMOC[cc] ./ (pp.q_AMOC * pp.AMOC_2010)
            end
        else
            # calculate CO2 release depending on AMOC strength
```

…you might laugh…



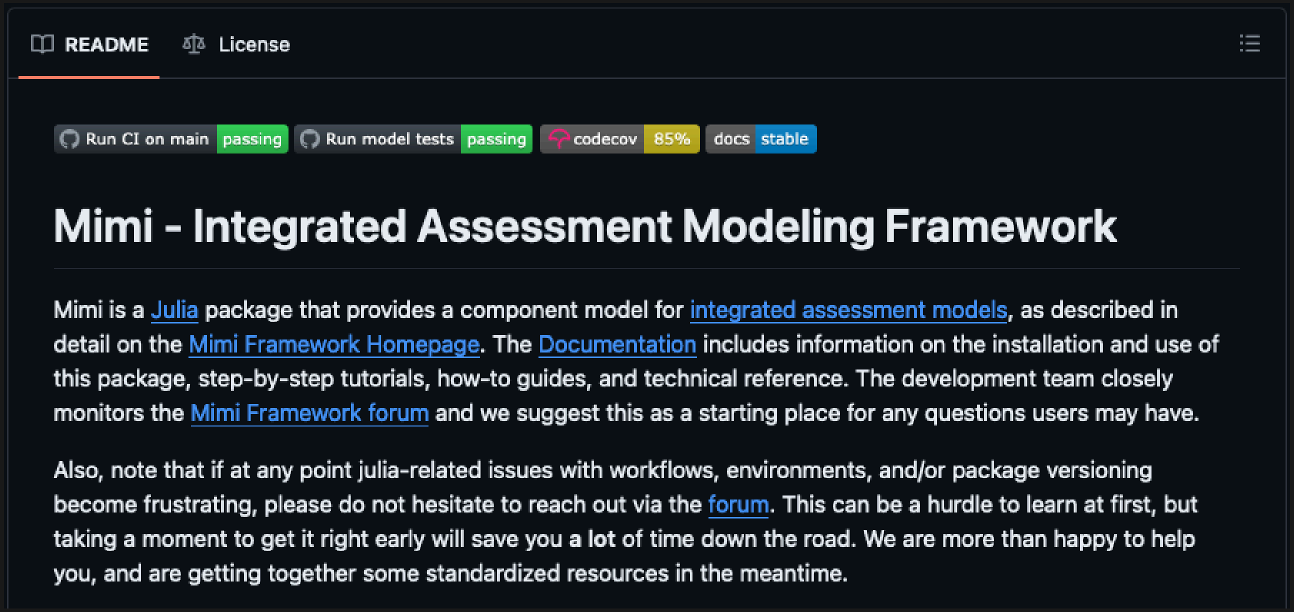
…is Excel

Also common: GAMS



- nonlinear optimisation
- fast and reliable solvers
- proprietary software
- restrictive UI and ecosystem

The open alternative: Mimi


README License

Run CI on main passing Run model tests passing codecov 85% docs stable

Mimi - Integrated Assessment Modeling Framework

Mimi is a Julia package that provides a component model for integrated assessment models, as described in detail on the Mimi Framework Homepage. The Documentation includes information on the installation and use of this package, step-by-step tutorials, how-to guides, and technical reference. The development team closely monitors the Mimi Framework forum and we suggest this as a starting place for any questions users may have.

Also, note that if at any point julia-related issues with workflows, environments, and/or package versioning become frustrating, please do not hesitate to reach out via the forum. This can be a hurdle to learn at first, but taking a moment to get it right early will save you **a lot** of time down the road. We are more than happy to help you, and are getting together some standardized resources in the meantime.

- package within Julia
- entirely open-source
- many IAMs and impact models available
- in principle modular and interoperable
- used by the EPA for US SCC calculations

julia

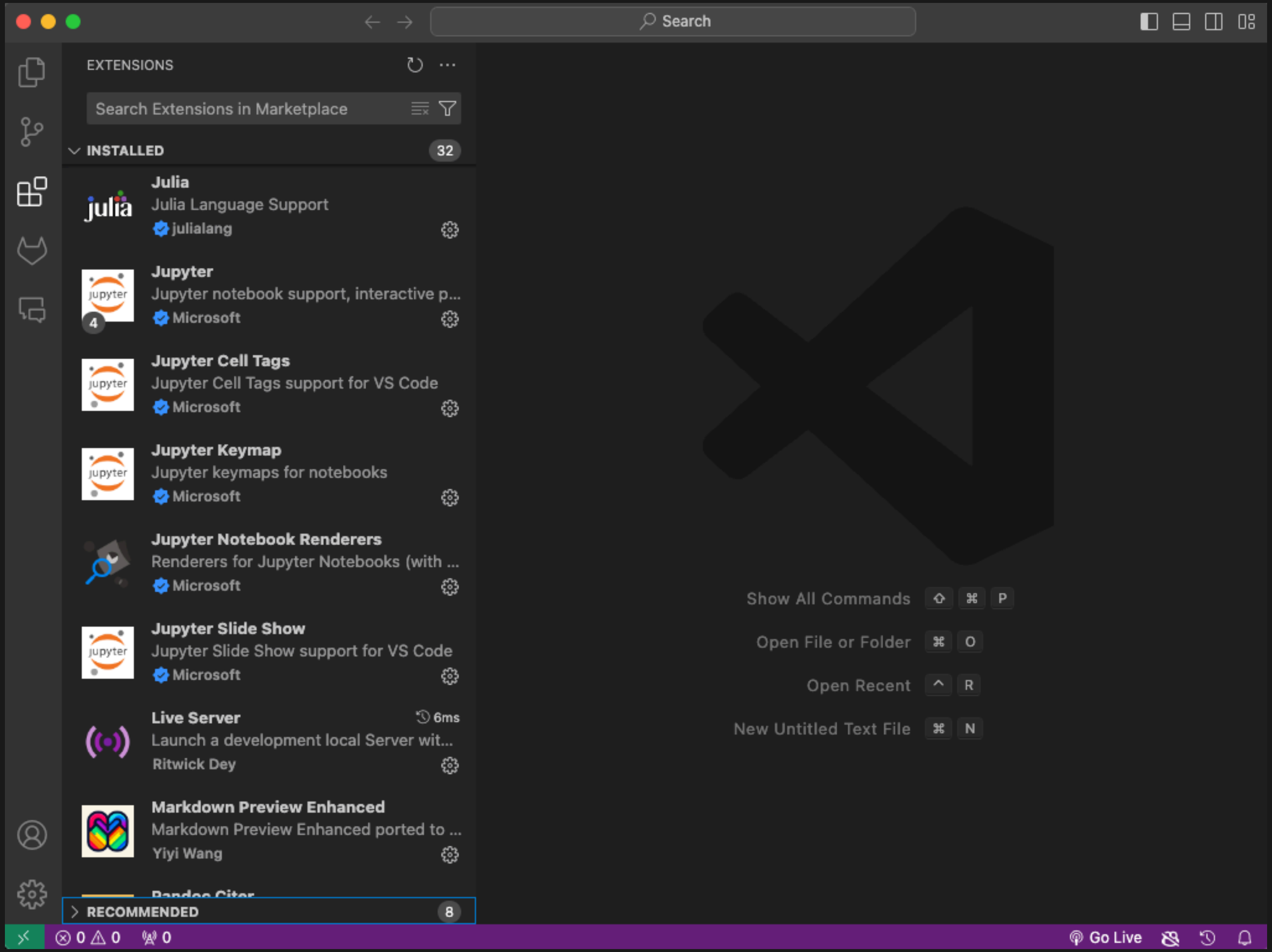
For the Mimi documentation click here.

☕ Break (& time for questions)

3. How to run an IAM with Mimi

```julia
@defcomp AMOC_emulator begin
    country = Index()

    # Variables
    I_AMOC = Variable{Bool}(index=[time])
    AMOC_decrease = Variable(index=[time], unit="Sv") # AMOC decrease (measured in Sv) starting in t=0
    deltaT_country_AMOC = Variable(index=[time, country], unit="degC")
    T_country_AMOC = Variable(index=[time, country], unit="degC")
    CO2_AMOC = Variable(index=[time], unit="GtCO2")
    cum_CO2_AMOC = Variable(index=[time], unit="GtCO2")

    # Parameters
    T_AT = Parameter(index=[time], unit="degC")
    f_AMOC = Parameter(index=[time])

    beta_AMOC = Parameter() # coefficient representing the sensitivity of AMOC to temperature
    DecreaseOffset = Parameter() # offset for AMOC decrease by 2010 in Sv (with how much weakening AMOC starts in 2010)
    eta_AMOC = Parameter() # coefficient representing the sensitivity of the carbon drawdown to AMOC decrease
    tau_AMOC = Variable(index=[country]) # coefficient representing the sensitivity of the temperature change to AMOC decrease
    I_tau = Parameter(default=1) # indicator variable for gamma_AMOC
    q_AMOC = Parameter() # coefficient indicating the relative amount of AMOC slowdown in a given calibration [%]

    # Delta_AMOC = Parameter(unit="year", default=35)
    max_deltaT_country_AMOC = Parameter(index=[country], unit="degC")

    T_country_base = Parameter(index=[time, country], unit="degC")

    T_AT_2010 = Parameter(default=0.854, unit="degC")
    AMOC_2010 = Parameter(default=18.0, unit="Sv")

    function run_timestep(pp, vv, dd, tt)

        # AMOC decrease (measured in Sv) starting in t=0 and assumed to be linear
        # this variable could also be directly calibrated to ESM output
        vv.AMOC_decrease[tt] = pp.DecreaseOffset + pp.beta_AMOC * (pp.T_AT[tt] - pp.T_AT_2010)
        vv.I_AMOC[tt] = 0

        if is_first(tt)

            # calculate CO2 release depending on AMOC strength
            # while not imposing a maximum CO2 release
            vv.CO2_AMOC[tt] = pp.eta_AMOC * vv.AMOC_decrease[tt]
            vv.cum_CO2_AMOC[tt] = vv.CO2_AMOC[tt]

            for cc in dd.country
                vv.tau_AMOC[cc] = pp.I_tau * pp.max_deltaT_country_AMOC[cc] ./ (pp.q_AMOC * pp.AMOC_2010)
            end
        else
            # calculate CO2 release depending on AMOC strength
```

Prerequisite 1: a functioning coding environment

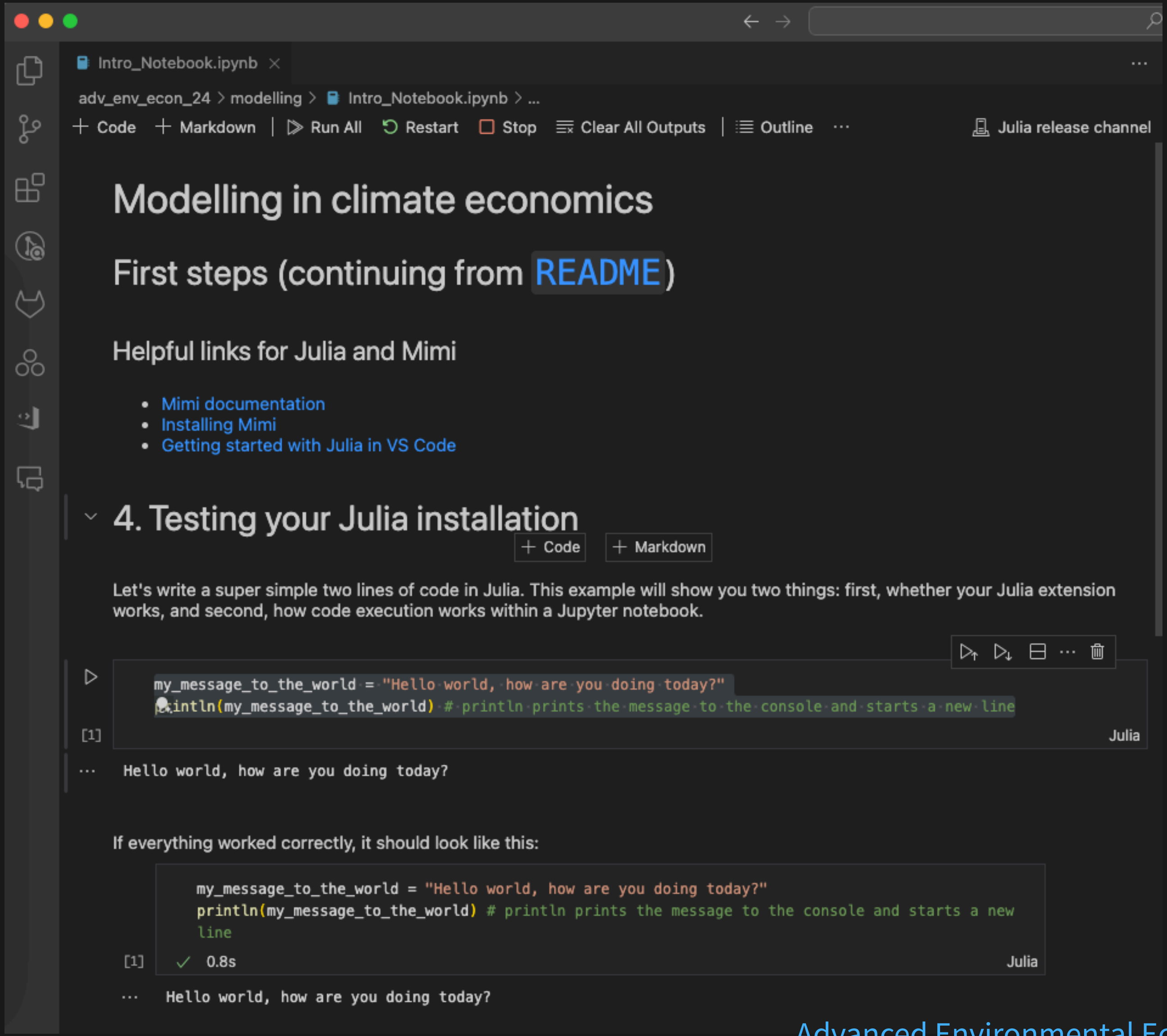


Do you all have this?

Can you open the lecture folder?

Can you update it?

Prerequisite 2: Julia and Jupyter installations

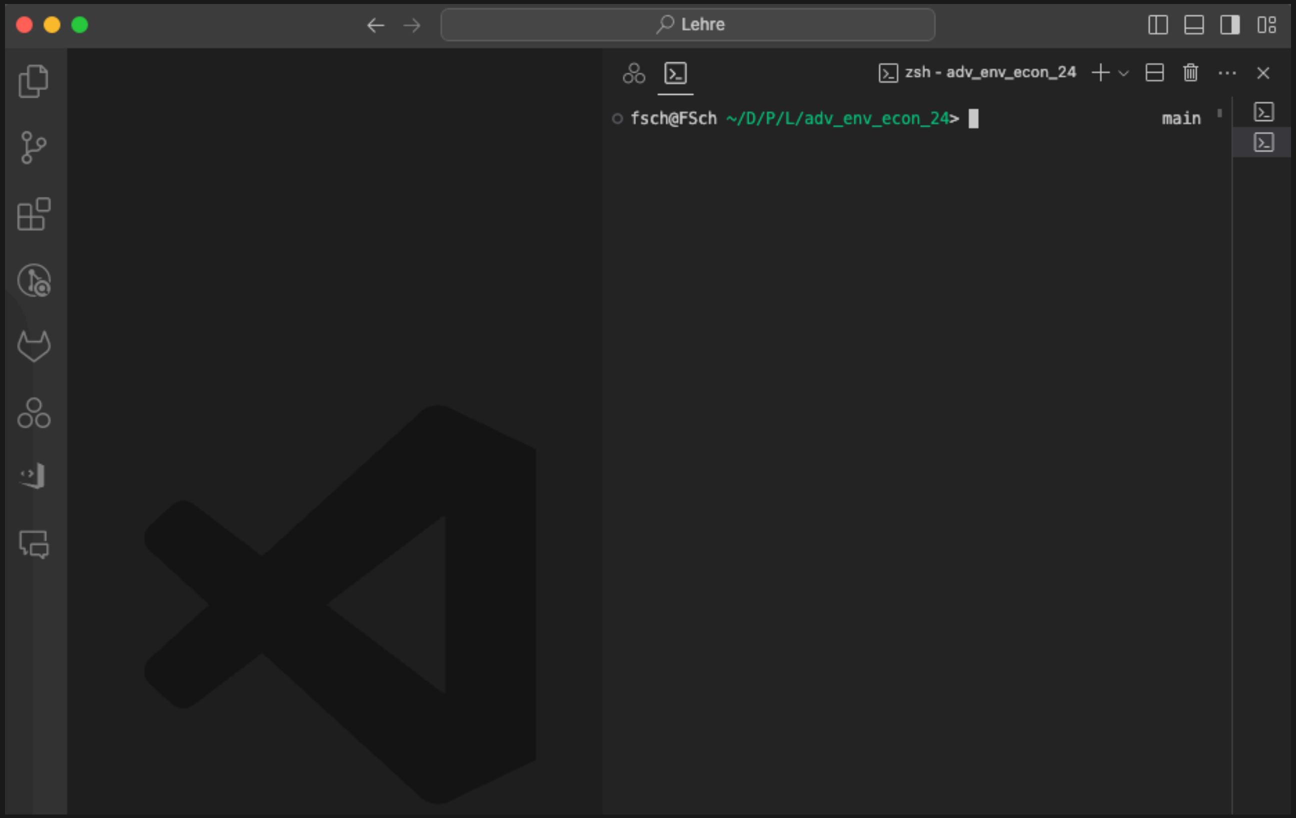


Can you open and render the `Intro_Notebook.ipynb`?

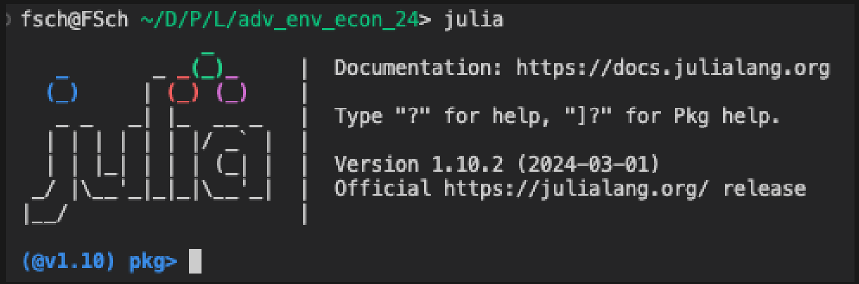
Does the Julia cell run?

Prerequisite 3: a proper Julia environment

Julia environment: a sandbox with a set of packages



1. Open the terminal within VS Code
2. Type `julia` and then type `]`

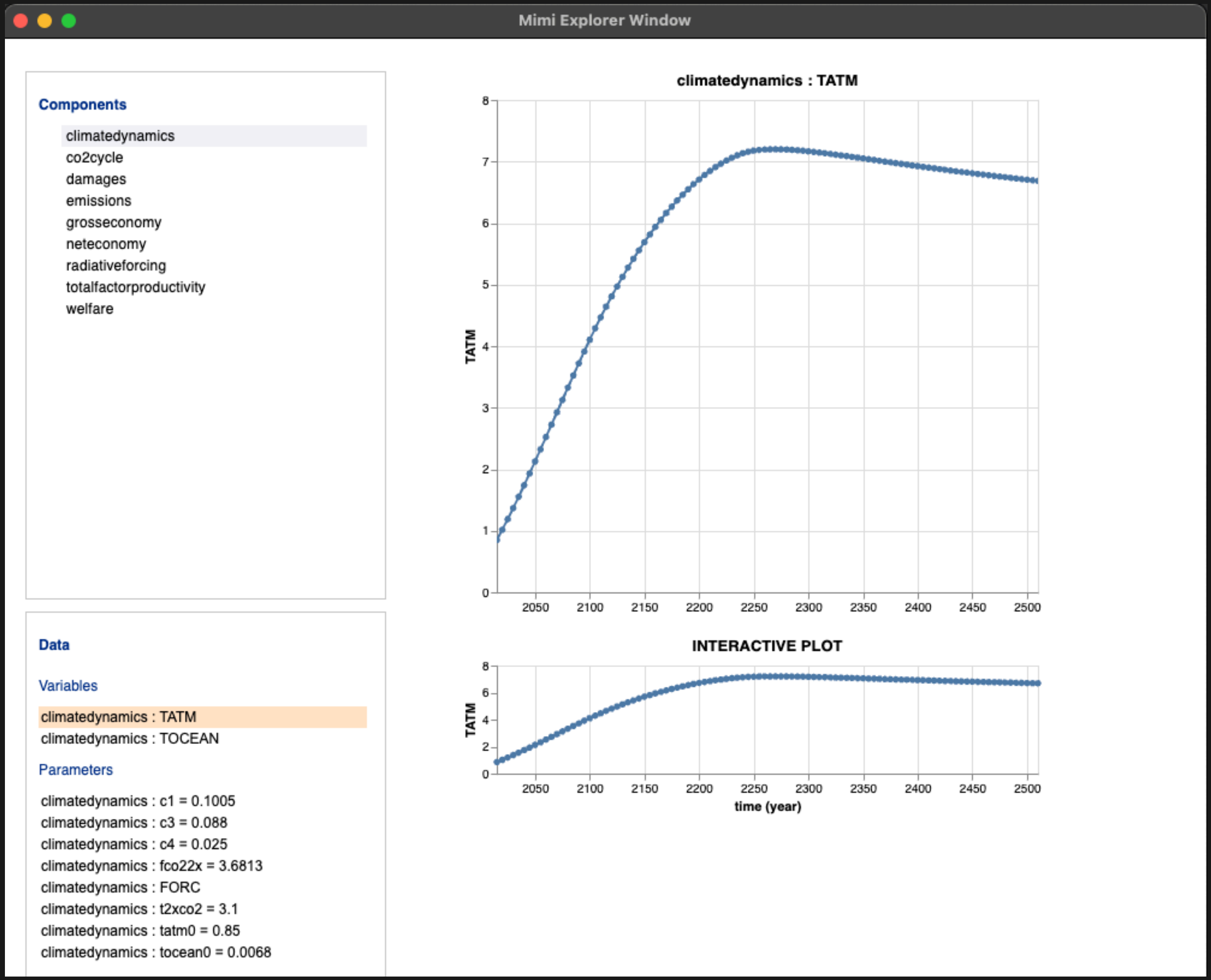


3. Installed packages: `status`
4. New environment in current folder: `activate .`

```
(adv_env_econ_24) pkg>
```

Running MimiDICE2016R2

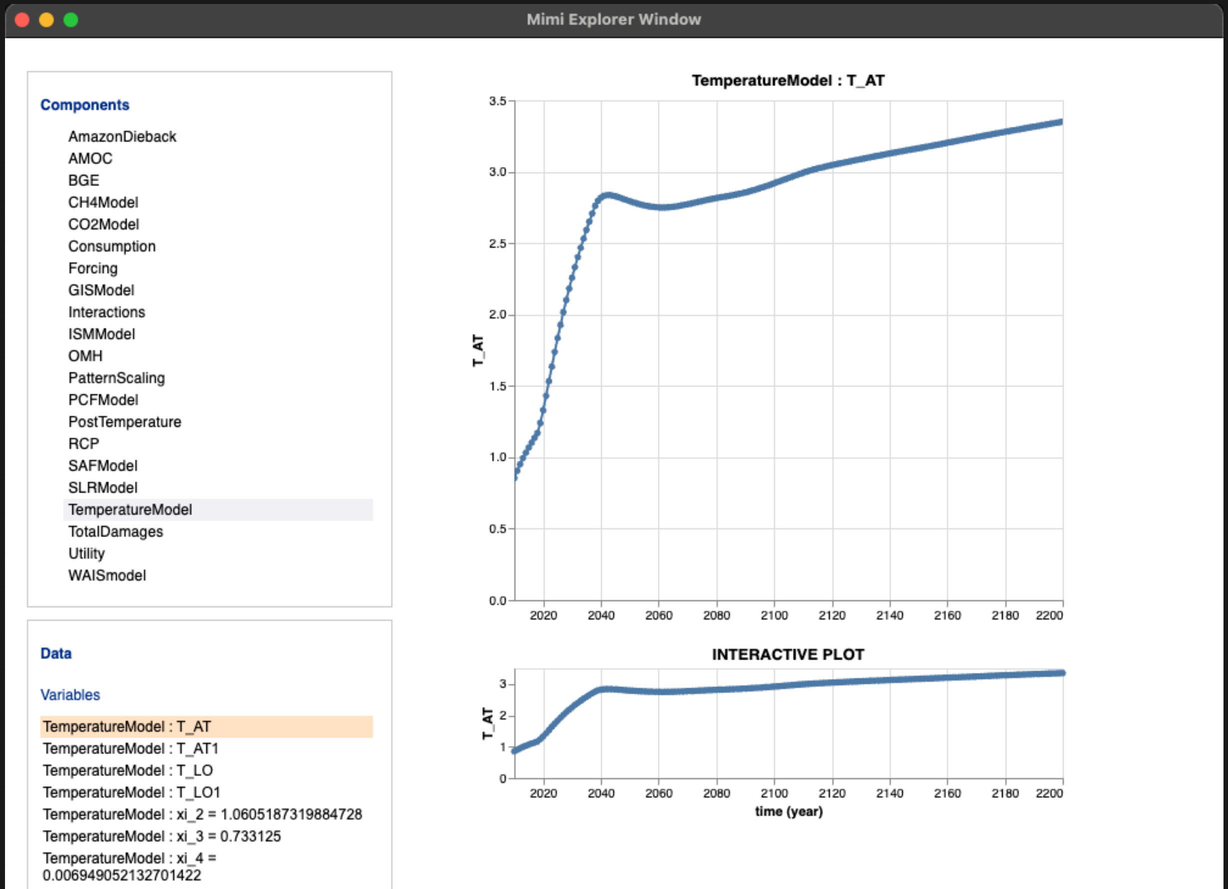
```
# import packages
using Mimi
using MimiDICE2016R2

# get model and build it
m = MimiDICE2016R2.get_model()
run(m)

# explore in graphical user interface
explore(m)
```



- MimiDICE2016R2 does not optimise
- hence: only simulation based on given parameter values
- for optimisation see here

Running META-2021

```
# import packages by calling main Julia fi
include("./META-2021/src/MimiMETA.jl")

# get model and build it
model = full_model(rcp="RCP4.5", ssp="SSP2'
run(model)

# explore in graphical user interface
explore(model)
```



- META not a package
- hence: clone and import file
- challenge: relative file paths

Next steps

- change parameters of a model
- change equations of a model
- check output and make some plots
- optimise a model